EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S32	24818788	@ad<"20030903"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 17:54
S33	7	S32 and "early miss"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 18:05
S34	17	S32 and miss$3 same replac$5 with algorithm with select$3 with "cache line"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 19:09
S35	48	S32 and output$4 with tag with predict$3 with way	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 19:26
S36	10	S32 and output$4 near3 tag with predict$3 with way	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 19:53
S37	0	S32 and activat$3 with (wordline (word adj line)) with ways with predict$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 19:56
S38	2	S32 and (wordline (word adj line)) with ways with predict$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 19:58

EAST Search History

S39	5520	way near3 predict$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S40	24818788	@ad<"20030903"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S41	47380	(decoder near3 address$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S42	660010	(set near3 (locations addresses values))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S43	72	S32 and S41 and S39 and S42	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S44	66163	(meier.in. nelson.in. shen.in.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S45	242461	(select$3 near3 set)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11

EAST Search History

S46	0	S32 and S41 same S45 same S42 same S39	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S47	5	S32 and S41 and S45 same S42 and S39	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S48	1	S32 and S41 same S45 same S42 and S39	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S49	8	S32 and S41 and (S42 same cache same way) and S39	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S50	0	S32 and S41 and S39 and "early miss"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/10 20:11
S51	3	S32 and S39 and (early near5 miss) and replac$5 with "cache line"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/10 20:11
S52	8	S32 and (early near5 miss) and replac$5 with "cache line"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/10 20:11

EAST Search History

S53	3	S32 and S41 and S39 and S42 and comparators with way with predict$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S54	34	S32 and S41 and S39 and S42 and comparators	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/10 20:11
S55	135	S32 and S45 same S42 and S39	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S56	3	S44 and (decoder with cache).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S57	1	S56 and (way adj prediction).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:11
S58	2	"20050050278".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:21
S59	5729	(711/128.ccls. 711/118.ccls. 711/133. ccls. 365/49.ccls. 365/50.ccls.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:21

EAST Search History

S60	27	S59 and (S33 S34 S35 S36 S38 S48 S43 S47 S49 S51 S52 S53 S54)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:27
S61	3	"7117290".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:28
S62	2	"5412787".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:28
S63	2	"5813031".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:28
S64	2	"5953748".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:29
S65	2	"6065091".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:29
S66	2	"6212602".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:30

EAST Search History

S67	2	"6687789".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:31
S68	114	"6115792".pn. or "6016533".pn. or "5845323".pn. or "5752069".pn. or "4912626".pn. or "5509119".pn. or "5091851".pn. or "5210845".pn. or "5148538".pn. or "5530958".pn. or "5559975".pn. or "6425055".pn. or "5918245".pn. or "6553477".pn. or "6418521".pn. or "5412787".pn. or "5893146".pn. or "5956746".pn. or "6240488".pn. or "5802594".pn. or "6016545".pn. or "5671444".pn. or "4914582".pn. or "5142634".pn. or "5235697".pn. or "5230068".pn. or "5327547".pn. or "5454117".pn. or "5485587".pn. or "5345569".pn. or "5521306".pn. or "5185868".pn. or "4984154".pn. or "4853840".pn. or "4764861".pn. or "5418922".pn. or "5283873".pn. or "5136697".pn. or "5053631".pn. or "5129067".pn. or "5058048".pn. or "4928223".pn. or "5764946".pn. or "5752069".pn. or "5701435".pn. or "5640532".pn. or "5423011".pn. or "5381533".pn. or "4943908".pn. or "5651125".pn. or "4044338".pn. or "4453212".pn. or "4807115".pn. or "4858105".pn. or "5226126".pn. or "5226130".pn. or "5619676".pn. or "5835951".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:51
S69	2	"5521306".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:53
S70	2	"5802594".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/10 20:53

Welcome United States Patent and Trademark Office

Search Session History BROWSE SEARCH IEEE XPLORE GUIDE

Wed, 12 Dec 2007, 5:27:54 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

[redacted]

Recent Search Queries

#1 (((decoder)<in>metadata) <and> ((generate)<in>metadata)) <and> ((way)<in>metadata)

#2 (((way)<in>metadata) <and> ((prediction)<in>metadata))

#3 (((predict)<in>metadata) <and> ((way)<in>metadata))<and> ((plurality)<in>metadata)

#4 (((decoder)<in>metadata) <and> ((way)<in>metadata))<and> ((predict)<in>metadata)

#5 (((set)<in>metadata) <and> ((way)<in>metadata))<and> ((access)<in>metadata)

#6 (((way)<in>metadata) <and> ((prediction)<in>metadata)) <and> ((generate)<in>metadata)

#7 (((way)<in>metadata) <and> ((prediction)<in>metadata)) <and> ((generate)<in>metadata)

#8 (((address)<in>metadata) <and> ((way)<in>metadata))<and> ((access)<in>metadata)

#9 (((early)<in>metadata) <and> ((miss)<in>metadata))<and> ((way)<in>metadata)

[redacted]

Terms used:
decoder decode address access cache set way generate prediction

Found **370** of **215,737**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 System-level power optimization: techniques and tools

Luca Benini, Giovanni de Micheli
April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2
Publisher: ACM Press

Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

2 A way-halting cache for low-energy high-performance systems

Chuanjun Zhang, Frank Vahid, Jun Yang, Walid Najjar
March 2005 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 2 Issue 1
Publisher: ACM Press

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, index terms

Caches contribute to much of a microprocessor system's power and energy consumption. Numerous new cache architectures, such as phased, pseudo-set-associative, way predicting, reactive-associative, way-shutdown, way-concatenating, and highly-associative, are intended to reduce power and/or energy, but they all impose some performance overhead. We have developed a new cache architecture, called a way-halting cache, that reduces energy further than previously mentioned architectures, while imposing ...

Keywords: Cache, dynamic optimization, embedded systems, low energy, low power

3 Power optimizations for cache memory: A way-halting cache for low-energy high-performance systems

Chuanjun Zhang, Frank Vahid, Jun Yang, Walid Najjar
August 2004 **Proceedings of the 2004 international symposium on Low power electronics and design ISLPED '04**

Publisher: ACM Press

Full text available: pdf(236.33 KB) Additional Information: full citation, abstract, references, citings, index terms

Caches contribute to much of a microprocessor system's power and energy consumption. We have developed a new cache architecture, called a way-halting cache, that reduces energy while imposing no performance overhead. Our way-halting cache is a four-way set-associative cache that stores the four lowest-order bits of all ways' tags into a fully associative memory, which we call the halt tag array. The lookup in the halt tag array is done in parallel with, and is no slower than, the set-index decod ...

Keywords: cache design, low power techniques

4 Balanced Cache: Reducing Conflict Misses of Direct-Mapped Caches

Chuanjun Zhang

May 2006 **ACM SIGARCH Computer Architecture News , Proceedings of the 33rd annual international symposium on Computer Architecture ISCA '06**, Volume 34 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(391.30 KB) Additional Information: full citation, abstract, index terms

Level one cache normally resides on a processor's critical path, which determines the clock frequency. Directmapped caches exhibit fast access time but poor hit rates compared with same sized set-associative caches due to nonuniform accesses to the cache sets, which generate more conflict misses in some sets while other sets are underutilized. We propose a technique to reduce the miss rate of direct mapped caches through balancing the accesses to cache sets. We increase the decoder length and th ...

5 A predictive decode filter cache for reducing power consumption in embedded processors

Weiyu Tang, Arun Kejariwal, Alexander V. Veidenbaum, Alexandru Nicolau

April 2007 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 12 Issue 2

Publisher: ACM Press

Full text available: pdf(267.90 KB) Additional Information: full citation, abstract, references, index terms

With advances in semiconductor technology, power management has increasingly become a very important design constraint in processor design. In embedded processors, instruction fetch and decode consume more than 40% of processor power. This calls for development of power minimization techniques for the fetch and decode stages of the processor pipeline. For this, filter cache has been proposed as an architectural extension for reducing the power consumption. A filter cache is placed betw ...

Keywords: Cache, embedded processors, power optimization

6 Optimizing the data cache performance of a software MPEG-2 video decoder

Peter Soderquist, Miriam Leeser

November 1997 **Proceedings of the fifth ACM international conference on Multimedia MULTIMEDIA '97**

Publisher: ACM Press

Full text available: pdf(1.76 MB) Additional Information: full citation, references, citings, index terms

7 Survey of code-size reduction methods

Árpád Beszédes, Rudolf Ferenc, Tibor Gyimóthy, André Dolenc, Konsta Karsisto

September 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 3
Publisher: ACM
Full text available: pdf(443.89 KB) Additional Information: full citation, abstract, references, citings, index terms

Program code compression is an emerging research activity that is having an impact in several production areas such as networking and embedded systems. This is because the reduced-sized code can have a positive impact on network traffic and embedded system costs such as memory requirements and power consumption. Although code-size reduction is a relatively new research area, numerous publications already exist on it. The methods published usually have different motivations and a variety of appli ...

Keywords: code compaction, code compression, method assessment, method evaluation

8 Early load address resolution via register tracking

Michael Bekerman, Adi Yoaz, Freddy Gabbay, Stephan Jourdan, Maxim Kalaev, Ronny Ronen

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2
Publisher: ACM
Full text available: pdf(143.17 KB) Additional Information: full citation, abstract, references, cited by, index terms

Higher microprocessor frequencies accentuate the performance cost of memory accesses. This is especially noticeable in the Intel's IA32 architecture where lack of registers results in increased number of memory accesses. This paper presents novel, non-speculative technique that partially hides the increasing load-to-use latency, by allowing the early issue of load instructions. Early load address resolution relies on register tracking to safely compute the addresses of memory refere ...

9 Compiler-driven cached code compression schemes for embedded ILP processors

Sergei Y. Larin, Thomas M. Conte

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture MICRO 32**
Publisher: IEEE Computer Society
Full text available: pdf(1.24 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

During the last 15 years, embedded systems have grown in complexity and performance to rival desktop systems. The architectures of these systems present unique challenges to processor microarchitecture, including instruction encoding and instruction fetch processes. This paper presents new techniques for reducing embedded system code size without reducing functionality. This approach is to extract the pipeline decoder logic for an embedded VLIW processor in software at system develo ...

10 Near-Optimal Precharging in High-Performance Nanoscale CMOS Caches

Se-Hyun Yang, Babak Falsafi

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture MICRO 36**
Publisher: IEEE Computer Society
Full text available: pdf(206.92 KB) Additional Information: full citation, abstract, citings, index terms

High-performance caches statically pull up the bit-linesin all cache subarrays to optimize cache accesslatency. Unfortunately, such an architecture results in asignificant waste of energy in nanoscale CMOS implementationsdue to high leakage and bitline discharge inthe unaccessed subarrays. Recent research advocatesbitline isolation to control precharging of individualsubarrays using bitline precharge devices. In this paper,we

carefully evaluate the energy and performancetrade-offs of bitline iso ...

11 GPGPU: general purpose computation on graphics hardware

David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

12 DSTRIDE: data-cache miss-address-based stride prefetching scheme for multimedia processors

Hariprakash. G, Achutharaman. R, Amos R. Omondi
January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture ACSAC '01**, Volume 23 Issue 4
Publisher: IEEE Computer Society, IEEE Computer Society Press
Full text available: pdf(928.14 KB) Publisher Site Additional Information: full citation, abstract, references

Prefetching reduces cache miss latency by moving data up in memory hierarchy before they are actually needed. Recent hardware-based stride prefetching techniques mostly rely on the processor pipeline information (e.g. program counter and branch prediction table) for prediction. Continuing developments in processor microarchitecture drastically change core pipeline design and require that existing hardware-based stride prefetching techniques be adapted to the evolving new processor architectures. ...

13 Instruction fetch and control flow: Branch predictor guided instruction decoding

Oliverio J. Santana, Ayose Falcón, Alex Ramirez, Mateo Valero
September 2006 **Proceedings of the 15th international conference on Parallel architectures and compilation techniques PACT '06**
Publisher: ACM Press
Full text available: pdf(369.16 KB) Additional Information: full citation, abstract, references, index terms

Fast instruction decoding is a challenge for the design of CISC microprocessors. A well-known solution to overcome this problem is using a trace cache. It stores and fetches already decoded instructions, avoiding the need for decoding them again. However, implementing a trace cache involves an important increase in the fetch architecture complexity.In this paper, we propose a novel decoding architecture that reduces the fetch engine implementation cost. Instead of using a special-purpose buffer ...

Keywords: branch predictor, complexity-effective, instruction decoding

14 Memory-wall: Boosting trace cache performance with nonhead miss speculation

Stevan Vlaovic, Edward S. Davidson
June 2002 **Proceedings of the 16th international conference on Supercomputing ICS '02**
Publisher: ACM Press
Full text available: pdf(179.52 KB) Additional Information: full citation, abstract, references, index terms

Trace caches are used to help dynamic branch prediction make multiple predic¿tions in a cycle by embedding some of the predictions in the trace. In this work, we evaluate a trace cache that is capable of delivering a trace consisting of a variable number of instructions via a linked list mechanism. We evaluate several schemes in the context of an x86 processor model that stores decoded instructions. By developing a new classification for trace cache accesses, we are able to target those misses t ...

Keywords: branch prediction, optimization, trace cache, x86

15 A highly configurable cache for low energy embedded systems

Chuanjun Zhang, Frank Vahid, Walid Najjar
May 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 2
Publisher: ACM Press

Full text available: pdf(714.89 KB) Additional Information: full citation, abstract, references, citings, index terms

Energy consumption is a major concern in many embedded computing systems. Several studies have shown that cache memories account for about 50% of the total energy consumed in these systems. The performance of a given cache architecture is determined, to a large degree, by the behavior of the application executing on the architecture. Desktop systems have to accommodate a very wide range of applications and therefore the cache architecture is usually set by the manufacturer as a best compr ...

Keywords: Cache, architecture tuning, configurable, embedded systems, low energy, low power, memory hierarchy, microprocessor

16 Reliable microarchitectures: Working with process variation aware caches

Madhu Mutyam, Vijaykrishnan Narayanan
April 2007 **Proceedings of the conference on Design, automation and test in Europe DATE '07**
Publisher: EDA Consortium

Full text available: pdf(383.73 KB) Additional Information: full citation, abstract, references

Deep-submicron designs have to take care of process variation effects as variations in critical process parameters result in large variations in access latencies of hardware components. This is severe in the case of memory components as minimum sized transistors are used in their design.

In this work, by considering on-chip data caches, we study the effect of access latency variations on performance. We discuss performance losses due to the worst-case design, wherein the entire cache o ...

17 Coupling compiler-enabled and conventional memory accessing for energy efficiency

Raksit Ashok, Saurabh Chheda, Csaba Andras Moritz
May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2
Publisher: ACM Press

Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, index terms

This article presents Cool-Mem, a family of memory system architectures that integrate conventional memory system mechanisms, energy-aware address translation, and compiler-enabled cache disambiguation techniques, to reduce energy consumption in general-purpose architectures. The solutions provided in this article leverage on interlayer tradeoffs between architecture, compiler, and operating system layers. Cool-Mem achieves power reduction by statically matching memory operations with energy-eff ...

Keywords: Energy efficiency, translation buffers, virtually addressed caches

18 Performance evaluation of a decoded instruction cache for variable instruction-length computers

Gideon Intrater, Ilan Spillinger

April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92**, Volume 20 Issue 2

Publisher: ACM

Full text available: pdf(832.44 KB) Additional Information: full citation, abstract, references, cited by, index terms

A Decoded INstruction Cache (DINC) serves as a buffer between the instruction decoder and the other instruction-pipeline stages. In this paper we explain how techniques that reduce the branch penalty based on such a cache, can improve CPU performance. We analyze the impact of some of the design parameters of DINCs on variable instruction-length computers, e.g., CISC machines. Our study indicates that tuning the mapping function of the instructions into the cache, can improve the ...

19 Course 4: State of the art in massive model visualization: Efficient data reduction and cache-coherent techniques toward real-time performance

Dave Kasik

August 2007 **ACM SIGGRAPH 2007 courses SIGGRAPH '07**

Publisher: ACM Press

Full text available: pdf(11.81 MB) Additional Information: full citation, references

20 Scalable Store-Load Forwarding via Store Queue Index Prediction

Tingting Sha, Milo M. K. Martin, Amir Roth

November 2005 **Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture MICRO 38**

Publisher: IEEE Computer Society

Full text available: pdf(306.61 KB) Publisher Site Additional Information: full citation, abstract, index terms

Conventional processors use a fully-associative store queue (SQ) to implement store-load forwarding. Associative search latency does not scale well to capacities and bandwidths required by wide-issue, large window processors. In this work, we improve SQ scalability by implementing store-load forwarding using speculative indexed access rather than associative search. Our design uses prediction to identify the single SQ entry from which each dynamic load is most likely to forward. When a load exec ...